PATENT APPLICATION
DOCKET NO.: 10010394-1

LISTING OF THE CLAIMS PER 37 C.F.R. §1.121

1. (Currently amended) A method of synchronizing processors in a simulated multiprocessor environment operable to execute a code portion to be debugged, comprising ~~the steps~~:

providing a synchronous breakpoint at a predetermined address location with respect to said code portion;

executing said code portion on said processors in said simulated multiprocessor environment from a fixed location; and

when a first processor of said processors encounters said synchronous breakpoint, terminating execution of said code portion on said first processor while continuing to execute said code portion on remaining processors.

2. (Currently amended) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 1, further comprising ~~the steps~~:

when a particular processor of said remaining processors in said simulated multiprocessor environment reaches said synchronous breakpoint, terminating execution of said code portion on said particular processor while continuing to execute said code portion in said simulated multiprocessor environment until each of said remaining processors has reached said synchronous breakpoint; and

thereafter, returning run control to a user associated with said simulated multiprocessor environment.

3. (Currently amended) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 1, wherein said ~~step of~~ providing a synchronous breakpoint comprises converting a standard breakpoint into said synchronous breakpoint.

4. (Original) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 3, wherein said standard breakpoint is converted into said synchronous breakpoint by issuing a BREAKPOINT SYNC_SET command.

5. (Original) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 4, further comprising the step of issuing a BREAKPOINT SYNC_RELEASE command in order to release said processors from said synchronous breakpoint.

6. (Original) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 4, wherein said step of executing said code portion on said processors is performed in a round robin fashion with respect to said processors.

7. (Original) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 4, wherein said code portion is selected from the group consisting of an application program, a firmware code portion, a booting sequence, a software tool, and an operating system, and further wherein said synchronous breakpoint comprises a breakpoint converted from a standard breakpoint.

8. (Original) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 1, wherein a debugger program is operated by said user associated with said simulated multiprocessor environment.

9. (Original) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 8, wherein said debugger program is integrated with said simulated multiprocessor environment.

10. (Currently amended) A method of debugging a code portion targeted for execution on a target hardware platform, comprising ~~the steps of~~:

providing an architectural simulator operable to simulate said target hardware platform, said architectural simulator having a debugger associated therewith, wherein said target hardware platform comprises a multiprocessor system;

initializing in said architectural simulator a list of processors included in said target hardware platform;

setting a synchronous breakpoint at a predetermined address location with respect to said code portion operable to be executed on said architectural simulator;

launching said code portion on said architectural simulator from a fixed location;

automatically stepping through said list of processors until each of said processors of said architectural simulator reaches said synchronous breakpoint; and

thereafter, returning program control to said debugger for performing a debug operation.

11. (Original) The method of debugging a code portion targeted for execution on a target hardware platform as set forth in claim 10, wherein said multiprocessor system comprises a symmetrical multiprocessor system.

12. (Original) The method of debugging a code portion targeted for execution on a target hardware platform as set forth in claim 10, wherein said multiprocessor system comprises an asymmetrical multiprocessor system.

13. (Original) The method of debugging a code portion targeted for execution on a target hardware platform as set forth in claim 10, wherein said multiprocessor system comprises a loosely coupled multiprocessor system.

14. (Original) The method of debugging a code portion targeted for execution on a target hardware platform as set forth in claim 10, wherein said multiprocessor system comprises a tightly coupled multiprocessor system.

15. (Original) The method of debugging a code portion targeted for execution on a target hardware platform as set forth in claim 10, wherein said step of setting a synchronous breakpoint at a predetermined address location is accomplished by converting a standard breakpoint into said synchronous breakpoint upon issuing a BREAKPOINT SYNC_SET command.

16. (Original) A system for debugging a code portion targeted for execution on a target hardware platform, comprising:

an architectural simulator operable to simulate said target hardware platform, said architectural simulator having a debugger associated therewith, wherein said target hardware platform comprises a multiprocessor system;

means for initializing in said architectural simulator a list of processors included in said target hardware platform;

means for setting a synchronous breakpoint at a predetermined address location with respect to said code portion operable to be executed on said architectural simulator;

means for launching said code portion on said architectural simulator from a fixed location;

means for automatically stepping through said list of processors until each of said processors of said architectural simulator reaches said synchronous breakpoint; and

means for returning program control to said debugger for performing a debug operation after said processors have reached said synchronous breakpoint.

17. (Original) The system for debugging a code portion targeted for execution on a target hardware platform as set forth in claim 16, wherein said multiprocessor system comprises a symmetrical multiprocessor system.

18. (Original) The system for debugging a code portion targeted for execution on a target hardware platform as set forth in claim 16, wherein said multiprocessor system comprises an asymmetrical multiprocessor system.

19. (Original) The system for debugging a code portion targeted for execution on a target hardware platform as set forth in claim 16, wherein said multiprocessor system comprises a loosely coupled multiprocessor system.

20. (Original) The system for debugging a code portion targeted for execution on a target hardware platform as set forth in claim 16, wherein said multiprocessor system comprises a tightly coupled multiprocessor system.

21. (Original) The system for debugging a code portion targeted for execution on a target hardware platform as set forth in claim 16, wherein said means for setting a synchronous breakpoint at a predetermined address location is operable to convert a standard breakpoint into said synchronous breakpoint by issuing a BREAKPOINT SYNC_SET command.

22. (New) A method of synchronizing processors in a simulated multiprocessor environment operable to execute a code portion to be debugged, comprising:

issuing a BREAKPOINT SYNC_SET command to convert a standard breakpoint into a synchronous breakpoint;

providing said synchronous breakpoint at a predetermined address location with respect to said code portion;

executing said code portion on said processors in said simulated multiprocessor environment from a fixed location; and

when a first processor of said processors encounters said synchronous breakpoint, terminating execution of said code portion on said first processor while continuing to execute said code portion on remaining processors.

23. (New) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 22, further comprising:

when a particular processor of said remaining processors in said simulated multiprocessor environment reaches said synchronous breakpoint, terminating execution of said code portion on said particular processor while continuing to execute said code portion in said simulated multiprocessor environment until each of said remaining processors has reached said synchronous breakpoint; and

thereafter, returning run control to a user associated with said simulated multiprocessor environment.

24. (New) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 22, further comprising issuing a BREAKPOINT SYNC_RELEASE command in order to release said processors from said synchronous breakpoint.

25. (New) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 22, wherein said executing of said code portion on said processors is performed in a round robin fashion with respect to said processors.

26. (New) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 22, wherein said code portion is selected from the group consisting of an application program, a firmware code portion, a booting sequence, a software tool, and an operating system.

27. (New) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 22, wherein a debugger program is operated by said user associated with said simulated multiprocessor environment.

28. (New) The method of synchronizing processors in a simulated multiprocessor environment as set forth in claim 27, wherein said debugger program is integrated with said simulated multiprocessor environment.

29. (New) A method of debugging a code portion targeted for execution on a target hardware platform, comprising:

providing an architectural simulator operable to simulate said target hardware platform, said architectural simulator having a debugger associated therewith, wherein said target hardware platform comprises a multiprocessor system;

initializing in said architectural simulator a list of processors included in said target hardware platform;

converting a standard breakpoint into a synchronous breakpoint upon issuing a BREAKPOINT SYNC_SET command;

setting said synchronous breakpoint at a predetermined address location with respect to said code portion operable to be executed on said architectural simulator;

launching said code portion on said architectural simulator from a fixed location;

automatically stepping through said list of processors until each of said processors of said architectural simulator reaches said synchronous breakpoint; and

thereafter, returning program control to said debugger for performing a debug operation.

30. (New) The method of debugging a code portion targeted for execution on a target hardware platform as set forth in claim 29, wherein said multiprocessor system comprises a symmetrical multiprocessor system.

31. (New) The method of debugging a code portion targeted for execution on a target hardware platform as set forth in claim 29, wherein said multiprocessor system comprises an asymmetrical multiprocessor system.

32. (New) The method of debugging a code portion targeted for execution on a target hardware platform as set forth in claim 29, wherein said multiprocessor system comprises a loosely coupled multiprocessor system.

33. (New) The method of debugging a code portion targeted for execution on a target hardware platform as set forth in claim 29, wherein said multiprocessor system comprises a tightly coupled multiprocessor system.

34. (New) A system for debugging a code portion targeted for execution on a target hardware platform, comprising:

an architectural simulator operable to simulate said target hardware platform, said architectural simulator having a debugger associated therewith, wherein said target hardware platform comprises a multiprocessor system;

means for initializing in said architectural simulator a list of processors included in said target hardware platform;

means for converting a standard breakpoint into a synchronous breakpoint upon issuing a BREAKPOINT SYNC_SET command;

means for setting said synchronous breakpoint at a predetermined address location with respect to said code portion operable to be executed on said architectural simulator;

means for launching said code portion on said architectural simulator from a fixed location;

means for automatically stepping through said list of processors until each of said processors of said architectural simulator reaches said synchronous breakpoint; and

means for returning program control to said debugger for performing a debug operation after said processors have reached said synchronous breakpoint.

35. (New) The system for debugging a code portion targeted for execution on a target hardware platform as set forth in claim 34, wherein said multiprocessor system comprises a symmetrical multiprocessor system.

36. (New) The system for debugging a code portion targeted for execution on a target hardware platform as set forth in claim 34, wherein said multiprocessor system comprises an asymmetrical multiprocessor system.

37. (New) The system for debugging a code portion targeted for execution on a target hardware platform as set forth in claim 34, wherein said multiprocessor system comprises a loosely coupled multiprocessor system.

38. (New) The system for debugging a code portion targeted for execution on a target hardware platform as set forth in claim 34, wherein said multiprocessor system comprises a tightly coupled multiprocessor system.